



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/670,073	09/26/2000	Richard M. Shupak	777.345US1	5749
27488	7590	01/31/2005	EXAMINER	
MICROSOFT CORPORATION C/O MERCHANT & GOULD, L.L.C. P.O. BOX 2903 MINNEAPOLIS, MN 55402-0903			BATES, KEVIN T	
			ART UNIT	PAPER NUMBER
			2155	

DATE MAILED: 01/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/670,073

Applicant(s)

SHUPAK ET AL.

Examiner

Kevin Bates

Art Unit

2155

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 November 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-63 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-63 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Response to Amendment

This Office Action is in response to a communication made on November 23, 2005.

Claims 1-63 are pending in this application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leblang (5649200) in view of Schmidt (4558413) (Previously cited).

Regarding claim 1, Leblang discloses a computer-readable medium having computer-executable instructions comprising: querying a first server for the location information using the request, wherein the location information includes information about a location of a second server (Column 5, lines 45 – 53; Column 8, lines 31 – 35), that comprises the update information (Column 6, lines 4 – 19); linking the first server to the second server (Column 8, lines 31 – 35); querying the second server for the update information (Column 11, lines 53 – 64); receiving the update information from the second server (Column 9, lines 41 – 45; Column 12, lines 8 – 17), and updating software based on the update information (Column 9, lines 41 – 45), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with

update information that is used to update the corresponding version of the software;
generating a request to obtain location information of the update information using the
unique identifier; and that the software version is identified by the unique identifier.
Schmidt teaches a system for find software updates, which includes assigning a value
to a unique identifier that is used to identify a version of software associated with an
executable (Column 9, lines 43 – 47), where in the value is associated with update
information that is used to update the corresponding version of the software (Column 9,
lines 43 – 47); generating a request to obtain location information of the update
information using the unique identifier; and that the software version is identified by the
unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of
ordinary skill in the art at the time the invention was made to use Schmidt's teachings of
using unique identifiers to identify and locate software versions in a distributed system
in Leblang's system in order to handle large programs and modules with multiple
versions (Column 1, lines 26 – 36).

Regarding claim 2, Leblang discloses that querying a first server further comprises providing a path to a look up HyperText Transfer Protocol (HTTP) symbol location server (Column 8, lines 43 – 51).

Regarding claim 3, Leblang discloses that querying a first server further comprises querying a Dynamic Host Configuration Protocol (DHCP) server and requesting Uniform Resource Identifiers (URIs) to query the second server for the update information (Column 6, lines 1 – 19).

Regarding claim 4, Leblang discloses that querying a first server further comprises querying a Domain Name System (DNS) server for a service (SRV) record identifying the second server to be queried (Column 8, lines 43 – 51).

Regarding claim 5, Leblang discloses that querying a first server further comprises querying a directory service for the location of the second server (Column 6, lines 51 – 61).

Regarding claim 6, Leblang discloses that querying a first server further comprises querying an Application Configuration, Access Protocol (ACAP) server for the location of the second server (Column 12, lines 56 – 67).

Regarding claim 7, Leblang discloses that querying a first server further comprises querying a Lightweight Directory Access Protocol (LDAP) server for the location of the second server (Column 8, lines 30 – 35; Column 11, lines 61 – 64).

Regarding claim 8, Leblang discloses a computer-readable medium having computer-executable instructions comprising: creating a path to a symbol location server (Column 14, lines 25 – 39); querying the symbol location server through the path for the symbols (Column 16, lines 31 – 44; lines 62 – 67); receiving the symbol from symbol location server through the path; and updating software associated with the local file using the symbol (Column 18, lines 25 – 32) but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and

Art Unit: 2155

that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 9, Leblang discloses that assigning a value further comprises extracting the unique identifier from an image header of the local file (Column 15, lines 47 – 51; Column 10, line 65 – Column 11, line 3).

Regarding claim 10, Leblang discloses that a value is not replicated between differing versions of the local file (Column 9, lines 47 – 57).

Regarding claim 11, Leblang discloses that receiving the symbol further comprises receiving a file comprising the symbol, wherein the file is stored in a local system memory (Column 13, line 65 – Column 14, line 7).

Regarding claim 12, Leblang discloses that querying the symbol location server for further comprises querying the symbol location server with a user customized query which extracts over a back end store (Column 16, lines 40 – 44).

Regarding claim 13, Leblang discloses a computer-readable medium having computer-executable instructions comprising: creating a path to the first server (Column 14, lines 25 – 39; Column 2, line 2); querying the first server through the path for the location information using the request (Column 2, lines 6 – 14); receiving the update information from the first server through the path; and updating software associated with the executable based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 14, Leblang discloses that receiving the update information further comprises receiving reference location associated the second server to access a file associated with the executable on the second server (Column 2, line 65 – Column 3, line 10).

Regarding claim 15, Leblang discloses the idea of querying the first server further comprises querying a server selected from a group consisting of a DHCP server, a DNS server, an ACAP server, and a LDAP server (Column 3, lines 31 – 40; Column 2, line 65 – Column 3, line 1; Column 8, lines 43 – 51; Column 12, lines 56 – 67) where the type of connection and query and action the user takes, allows the system to take different action with the first server/ distributed system.

Regarding claim 16, Leblang discloses that querying the first server further comprises querying a set of servers in parallel (Column 16, lines 6 – 9) where the script allows the search of may objects/files at once and it also allows different types of searching for those files such as certain version location or query into the metadata.

Regarding claim 17, Leblang discloses that querying the first server further comprises querying a set of servers in a serial order (Column 2, lines 65 – Column 3, line 4; Column 16, lines 62 – 67) because in order to connect all the objects and locate all related information to those objects the system needs to transverse all paths and types of location techniques in the system, such as first locating the object and the pathnames to that objects versions, then looking in the object directory for metadata leading to hyperlinks for other objects which are merged or related.

Regarding claim 18, Leblang discloses that querying the first server further comprises packaging information extracted from the executable into a HyperText Transfer Protocol (HTTP) request and sending the HTTP request to first server (Column 9, lines 47 – 61).

Regarding claim 19, Leblang discloses a computer-readable medium having computer-executable instructions comprising: querying a first server for the location information using the request, wherein the location information includes information about a location of a second server that comprises the update information (Column 2, lines 53 – Column 3, line 9); receiving the location information from the first server; creating a path to the second server based on the type of the update information (Column 2, lines 53 – Column 3, line 9; Column 16, lines 31 – 44); querying the second server through the path for the update information associated with the executable using a syntax based on the location information received for the second server (Column 16, lines 40 – 44; Column 16, line 62 – Column 17, line 5); and updating software associated with the executable based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60) but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to

Art Unit: 2155

identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 20, Leblang discloses that querying a first server further comprises querying the first server using metadata associated with the executable (Column 15, lines 46 – 50).

Regarding claim 21, Leblang discloses that querying the second server further comprises querying the second server using metadata associated with the executable (Column 16, lines 40 – 44; lines 63 – 67).

Regarding claim 22, Leblang discloses that the metadata comprises metadata for a number of debug files (Column 17, lines 2 – 10).

Regarding claim 23, Leblang discloses that the metadata comprises metadata for a number of source files (Column 2, lines 23 – 29).

Regarding claim 24, Leblang discloses that querying the second server further comprises querying the second server for symbols associated with the executable (Column 2, lines 65 – Column 3, line 4).

Regarding claim 25, Leblang discloses that querying the second server further comprises querying the second server for regression analysis data associated with the executable (Column 5, line 61 – Column 6, line 3).

Regarding claim 26, Leblang discloses that querying the second server further comprises querying the second server for performance analysis data associated with the executable (Column 5, line 61 – Column 6, line 3).

Regarding claim 27, Leblang discloses that querying the second server further comprises querying the second server for source code associated with the executable (Column 5, line 61 – Column 6, line 3).

Regarding claim 28, Leblang discloses that querying the second server further comprising receiving files comprising the update information associated with the executable file (Column 5, line 61 – Column 6, line 3).

Regarding claim 29, Leblang discloses a computer-readable medium having computer-executable instructions for updating a software program associated with an executable file comprising: packaging metadata extracted from the executable file into an HTTP request (Column 2, lines 24 – 52); creating a path to a locator server (Column 14, lines 25 – 39; Column 2, line 2), wherein the locator server comprises location information for a server on which the update information is located (Column 2, line 65 – Column 3, line 4); sending through the path the HTTP request to the locator servers; receiving the update information from the locator server through the path (Column 2, line 65 – Column 3, line 4); and updating the software program based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly

Art Unit: 2155

indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 30, Leblang discloses that packaging metadata further comprising packaging metadata to locate an updated version of the executable file (Column 11, lines 53 – 64).

Regarding claim 31, Leblang discloses that packaging metadata further comprises packaging metadata for locating a debug file associated with the executable file (Column 17, lines 2 – 10).

Regarding claim 32, Leblang discloses that packaging metadata further comprises packaging metadata to locate a specific build version of the executable file (Column 9, lines 39 – 57).

Regarding claim 33, Leblang discloses that receiving the update information further comprises receiving an HTTP redirect (Column 2, line 65 – Column 3, line 1).

Regarding claim 34, Leblang discloses that receiving the update information further comprises receiving a location of the server on which the update information is located, and querying the server with the unique identifier for the update information (Column 7, lines 54 – 61; Column 17, lines 22 – 29).

Regarding claim 35, Leblang discloses that querying the server further comprises providing a qualifier (Column 15, lines 46 – 51).

Regarding claim 36, Leblang discloses a computerized system, comprising: a first server comprising location information for update information associated with a local file (Column 2, lines 53 – 64; Column 11, lines 53 – 60); a second server comprising the update information, wherein the first server is linked to the second server through a path that is created based on the value (Column 2, line 65 – Column 3, line 4), and further wherein the first server receives the update information from the second server through the path (Column 15, line 56 – Column 16, line 9); and a computer comprising the local file, wherein the first server provides the update information to the computer such that software associated with the local file is updated based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a

Art Unit: 2155

version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 37, Leblang discloses that the update information comprises debug information (Column 17, lines 2 – 10).

Regarding claim 38, Leblang discloses that the update information comprises solution access information (Column 5, line 61 – Column 6, line 5).

Regarding claim 39, Leblang discloses that the computer reads the update information from the second server (Column 8, lines 31 – 35).

Regarding claim 40, Leblang discloses that the first server comprises a HyperText Transfer Protocol (HTTP) server (Column 8, lines 43 – 51).

Regarding claim 41, Leblang discloses that the HTTP server comprises a Dynamic Host Configuration Protocol (DHCP) server having Uniform Resource Identifiers (URIs) for querying the second server (Column 6, lines 1 – 19).

Regarding claim 42, Leblang discloses that the HTTP server comprises a Domain Name System (DNS) server having a service (SRV) record for identifying the second server (Column 8, lines 43 – 51).

Regarding claim 43, Leblang discloses that the HTTP server comprises a directory service for providing the location information for the update information to the computer (Column 6, lines 51 – 61).

Regarding claim 44, Leblang disclose that the first server comprises an Application Configuration Access Protocol (ACAP) server (Column 12, lines 56 – 67).

Regarding claim 45, Leblang discloses that the first server comprises a Lightweight Directory Access Protocol (LDAP) server (Column 8, lines 30 – 35; Column 11, lines 61 – 64).

Regarding claim 46, Leblang does not explicitly indicate that the computer is networked to the first and the second servers over the Internet. Leblang discloses that the system is a distributed network, with servers and workstations (Column 6, lines 4 – 9) and that it uses pathnames to locate servers and information (Column 2, lines 65 – 66). It would have been obvious to one of ordinary skill in the art at the time the invention was made to encourage the scalability of the system to enable it to run over a

wider area network such as the Internet to allow a larger more geographically disperse development team to operate cooperatively in this system (Column 6, lines 10 – 26).

Regarding claim 47, Leblang discloses a computerized system comprising: a first server comprising location information for update information associated with an executable file (Column 5, lines 45 – 53; Column 8, lines 31 – 35; Column 2, lines 53 – 64; Column 11, lines 53 – 60); a second server comprising the update information (Column 6, lines 4 – 10), wherein the first server is linked to the second server through a path that is created based on the value (Column 2, line 67 – Column 3, line 4); and a computer comprising the executable file, wherein the first server is adapted to provide the computer with the location information (Column 2, line 67 – Column 3, line 4), and further wherein the computer uses the location information to query the second server through the path for the update information (Column 8, lines 43 – 51) such that software is updated based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of

Art Unit: 2155

the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 48, Leblang discloses the first server is selected from a group consisting of a DHCP server, a DNS server, an ACAP server, and a LDAP server (Column 3, lines 31 – 40; Column 2, line 65 – Column 3, line 1; Column 8, lines 43 – 51; Column 12, lines 56 – 67) where the type of connection and query and action the user takes, allows the system to take different action with the first server/ distributed system.

Regarding claim 49, Leblang discloses that the computer is configured to query a hierarchy of first servers in a serial order (Column 2, lines 65 – Column 3, line 4; Column 16, lines 62 – 67) because in order to connect all the objects and locate all related information to those objects the system needs to transverse all paths and types of location techniques in the system, such as first locating the object and the pathnames to that objects versions, then looking in the object directory for metadata leading to hyperlinks for other objects which are merged or related.

Regarding claim 50, Leblang discloses that the computer is configured to query a hierarchy of first servers in a parallel (Column 16, lines 6 – 9) where the script allows

the search of many objects/files at once and it also allows different types of searching for those files such as certain version location or query into the metadata.

Regarding claim 51, Leblang does not explicitly indicate that the update information comprises solution access information (Column 5, line 61 – Column 6, line 5).

Regarding claim 52, Leblang does not explicitly indicate that the computer is configured to query the second server, in an HTTP request format (Column 9, lines 47 – 61), for the update information using a qualifier associated with the executable file (Column 15, lines 46 – 51).

Regarding claim 53, Leblang discloses that the query to the second server is performed using metadata extracted from the executable file (Column 2; lines 24 – 52).

Regarding claim 54, Leblang discloses that the metadata extracted from the executable file comprising metadata for a debug file associated with the executable file (Column 17, lines 2 – 10).

Regarding claim 55, Leblang does not explicitly indicate that the metadata extracted from the executable file comprises metadata associated with regression analysis data for the executable file (Column 5, line 61 – Column 6, line 3).

Regarding claim 56, Leblang discloses that a computer readable medium having computer executable instructions to cause a computing system to perform a method for updating software associated with an executable file (Column 5, lines 45 – 53; Column 8, lines 31 – 35; Column 2, lines 53 – 64; Column 11, lines 53 – 60), comprising: creating a path for a lookup server to a server having update information based on the

Art Unit: 2155

value (Column 14, lines 25 – 39; Column 2, line 2); using the lookup server to identify location information for the server having the update information based on metadata extracted from the executable file; packaging an HTTP query for retrieving the update information through the path (Column 2, line 67 – Column 3, line 4; Column 8, lines 31 – 35; Column 16, lines 31 – 44; lines 62 – 67); retrieving the update information; and updating the software associated with the executable file based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system

in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 57, Leblang discloses that using the lookup server further comprises providing a response to a requesting client from the lookup server (Column 16, lines 40 – 44; Column 17, lines 50 – 57).

Regarding claim 58, Leblang discloses that providing a response further comprises forwarding the location information to the requesting client as an HTTP redirect (Column 8, lines 31 – 35).

Regarding claim 59, Leblang discloses a method for updating software associated with a local file comprising: packaging metadata extracted from the local file into an HTTP request to obtain the update information (Column 2, lines 24 – 52); sending the HTTP request to a locator server; receiving location information back from the locator server (Column 8, lines 31 – 35; Column 16, lines 40 – 44; Column 17, lines 50 – 57); packaging an HTTP query for retrieving the update information based on location information (Column 8, lines 43 – 51); and updating the software associated with the local file based on the information associated with the local file (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find

software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Regarding claim 60, Leblang does not explicitly indicate that packaging an HTTP query further comprises qualifying the HTTP query to select a specific file version from the update information (Column 8, lines 43 – 51).

Regarding claim 61, Leblang does not explicitly indicate that qualifying the HTTP query further comprises qualifying the HTTP query to select an updated file version associated with the local file (Column 11, lines 1 – 3)

Regarding claim 62, Leblang does not explicitly indicate that qualifying the HTTP query further comprises qualifying the HTTP query to select a specific debug file associated with the local file (Column 17, lines 1 – 9).

Regarding claim 63, Leblang discloses a server architecture, comprising; a first server; a second server linked to the first server based on a value (Column 5, lines 45 – 53; Column 8, lines 31 – 35; Column 2, lines 53 – 64; Column 11, lines 53 –

Art Unit: 2155

60), wherein the second server comprises update information associated with the executable file (Column 6, lines 4 – 10); means for interpreting metadata associated with an executable file received by the first server from a remote client (Column 16, lines 40 – 44); means for redirecting the remote client to the second server wherein the second server is adapted to interpret a query from the remote client for retrieving the update information (Column 8, lines 31 – 35; lines 43 – 51); and means for updating software associated with the executable based on the update information (Column 2, lines 53 – 64; Column 11, lines 53 – 60), but does not explicitly indicate assigning a value to a unique identifier that is used to identify a version of software associated with an executable, where in the value is associated with update information that is used to update the corresponding version of the software; generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier. Schmidt teaches a system for find software updates, which includes assigning a value to a unique identifier that is used to identify a version of software associated with an executable (Column 9, lines 43 – 47), where in the value is associated with update information that is used to update the corresponding version of the software (Column 9, lines 43 – 47); generating a request to obtain location information of the update information using the unique identifier; and that the software version is identified by the unique identifier (Column 9, lines 43 – 59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Schmidt's teachings of using unique identifiers to identify and locate

Art Unit: 2155

software versions in a distributed system in Leblang's system in order to handle large programs and modules with multiple versions (Column 1, lines 26 – 36).

Response to Arguments

Applicant's arguments with respect to claims 1-63 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin Bates whose telephone number is (571) 272-3980. The examiner can normally be reached on 8 am - 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571) 272-3978. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KB

KB
January 27, 2005



**HOSAIN ALAM
SUPERVISORY PATENT EXAMINER**